1. Introduction

The work of the Stanford Artificial Intelligence Project is charted in Figure 1.

In this introduction we shall make a few remarks about the outline as a whole, then comment on the status of the subdivisions and our plans for future work in them. More detailed descriptions and plans will be given later in the proposal.

This is not the place for a full discussion of the present state of research in artificial intelligence. Instead, we shall make a few statements about the general situation into which this proposal fits.

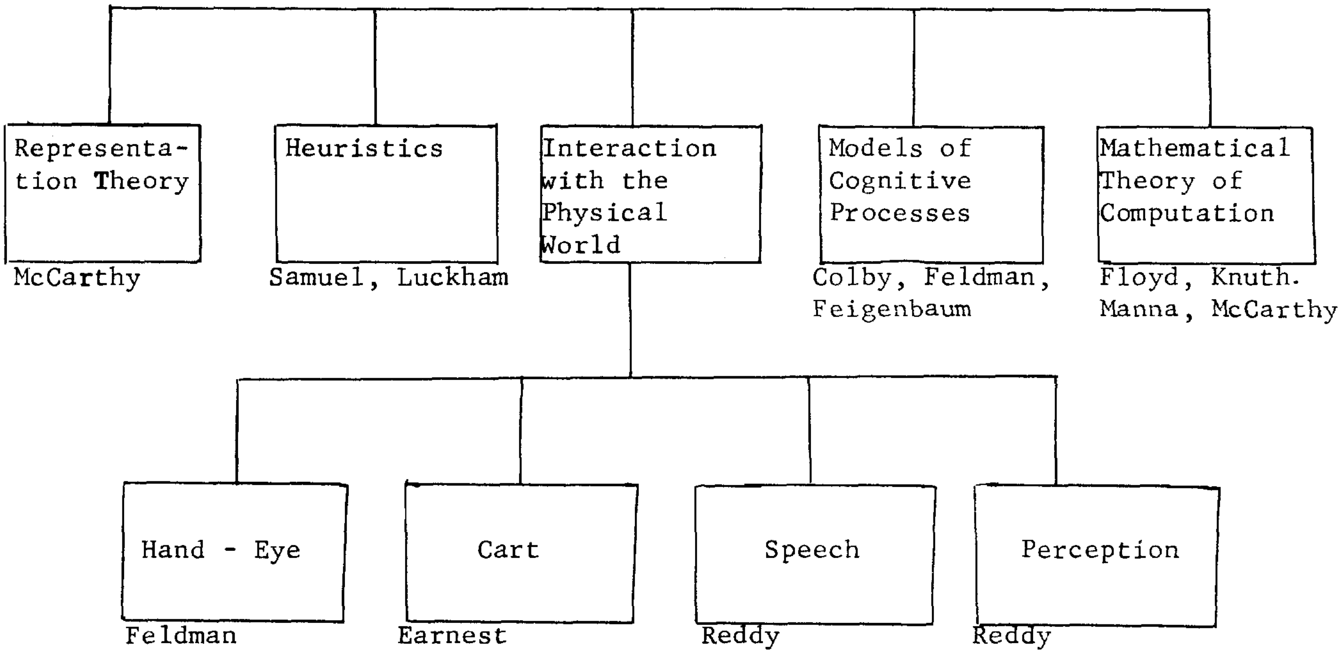
1. Artificial intelligence is the experimental and theoretical study of perceptual and intellectual processes using computers. Its ultimate goal is to understand these processes well enough to make a computer perceive, understand and act in ways now only possible for humans.

2. The information for this study comes in part from observation of human behavior, including self-observation, but mainly from experiments with programs designed to solve problems chosen to require the intellectual processes under study.

3. This understanding is at present in a very preliminary state, no one can say how long it will take to reach the understanding required to duplicate human intellectual performance because there are fundamental discoveries yet to be made.

4. Nevertheless, progress in identifying and duplicating intellectual mechanisms is being made and the range of problems that computers can be made to solve is increasing.

Figure 1. Structure of the Stanford Artificial Intelligence Project



5. Many blind alleys have been and are being followed and many mistakes are being made. Nevertheless, a body of fundamental knowledge is accumulating.

6. An important limitation up to the present has been the lack of well-trained and well-motivated scientists, working on artificial intelligence. The graduate program in computer science at Stanford has begun to change this situation.

7. Although full solution of the artificial intelligence problem is unpredictably far off, the understanding so far achieved has important potential practical applications. The development of these applications is worth undertaking.

We have divided our work in artificial intelligence into four categories: epistemology (representation theory), heuristics, interaction with the physical world, and models of cognitive processes.

The identification of the epistemological and heuristic parts of the artificial intelligence problems as separate entities is a result of work of the last few years, mainly by John McCarthy. The epistemological part is to choose a suitable representation for situations and the rules that describe how situations change. This description must be general enough to cover all problem solving situations, and even more important, it must be able to express all likely states of knowledge of the situation and the rules by which it changes spontaneously or by the actions of the problem solver.

The heuristic part of the artificial intelligence problem is to devise methods that, starting from a suitable representation of the information, will express explicitly the actions that have to be performed e.g., find the right chess move or the right next step in a proof. Work in heuristics has been carried on in many places for a number of years. Here, Samuel is leading work applying learning methods and Luckham is leading work in theorem proving by computer.

Samuel's new version of the checker program with learning is working and methods from that program are now being applied to learning in speech recognition. Luckham has been able to include in his resolution program almost all the methods found useful by others together with new methods of his own. Details on the results and future plans of the work are given in Section 5, below.

Much of our work involves modeling cognitive processes in one way or another, but a concentrated attack on this is through Feigenbaum's and Lederberg's Heuristic Dendral Project. This involves getting chemists to express their rules of reasonableness for the structure of organic compounds in a way that can be used by the computer program. Colby's work on belief systems and Feldman's language research are also aimed at developing cognitive models.

The largest area of work of the Stanford Artificial Intelligence Project both in terms of people and money, has been computer interaction with the physical world.

This work has gone slower than was anticipated when the project started. There are two reasons for this. First, it was more difficult, expensive, and time-consuming than expected to get a time-shared facility capable of both real-time and ordinary time-sharing. This task is substantially complete; all but reliability problems appear to be solved.

The second problem is that it has taken a long time to develop a group of good computer scientists whose main interest is the vision problem. The situation has greatly improved since the major hardware problems have been solved and since Professor Jerome Feldman has made hand-eye his major area of concentration. Now ten graduate students (Ruzena Bajcsyova, Gil Falk, Gunnar Grape, Lou Paul, Irwin Sobel, Jay Tennenbaum, Dave VanVoorhis, Bruce Baumgart, Rod Schmidt and Jack Buchanan) are doing research in this and other perception areas that should lead to theses.

Finally, in our work on the mathematical theory of computation, substantial results have been obtained by McCarthy and his students and more recently by Manna and Pnueli. The Stanford work in this area has been further strengthened by the addition of Robert Floyd and Donald Knuth to the faculty.

The balance of this proposal is divided into sections that discuss plans in each research area. The current research staff is given here for each topic.

Theory

Representation Theory

Fred Goldstein

Prof. John McCarthy

Mathematical Theory of Computation

Prof. Robert Floyd

Prof. Donald Knuth

Lockwood Morris

Steven Ness

Prof. Zohar Manna

Prof. John McCarthy

Visual Perception and Control

Hand-Eye Systems

Gilbert Falk
Prof. Jerome Feldman
Gunnar Grape
Richard Paul
Karl Pingle
Gerald Shapiro
Irwin Sobel
Jay Tenenbaum

Visual Perception

Ruzena Bajcsyova
Dr. Manfred Hueckel
Michael Kelly
Prof. D. Raj Reddy
David VanVoorhis

Visual Control of a Vehicle

Bruce Baumgart
Jack Buchanan
Lester Earnest
Rodney Schmidt

Speech Recognition

Dr. James Beauchamp
Lee Erman
Gary Goodman
Richard Neely
Stephen Plant
Prof. D. Raj Reddy

Heuristic Search

Machine Learning

Dr. Morton Astrahan
Johathan Ryder
Dr. Arthur Samuel
Joseph Siberz
Dr. George White

Theorem Proving

John Allen
Dr. David Luckham
Dr. Nils Nilsson

Models of Cognitive Processes

Heuristic Dendral

Prof. Malcolm Bersohn

Dennis Brown

Dr. Bruce Buchanan

Allan Delfino

Prof. Carl Djerassi

Dr. Allan Duffield

Isu Fang

Prof. Edward Feigenbaum

Dr. Gustav Schroll

Georgia Sutherland

Language Research

Prof. Jerome Feldman

Stephen Reder

Higher Mental Functions

Dr. Kenneth Colby

Dr. Franklin Hilf

Dr. Roger Schank

David Smith

Lawrence Tesler

Sylvia Weber

2. Theory

2.1 Representation Theory

Our recent theoretical work on artificial intelligence has led us to divide the subject into two parts: representation theory and heuristics. This division arises as follows.

When we try to make a computer program that solves a certain class of problems, our first task is to decide what information is involved in stating the problem and is available to help in its solution. Next we must decide how this information is to be represented in the memory of the computer. Only then can we choose the algorithms for manipulating this information to solve our problem. Representation theory deals with what information we need and how it is represented in the computer. Heuristics is concerned with the structure of the problem solving algorithms.

In the past, work in artificial intelligence has been content with a rather perfunctory approach to representations. A representation is chosen rather quickly for a class of problems and then all attention is turned to devising, programming and testing heuristics. The trouble with this approach is that the resulting programs lack generality and are not readily modifiable to attack new classes of problems.

The first goal of representation theory is to devise a general way of representing information in the computer. It should be capable of representing any state of partial information that a person might have about a problem and the general information necessary to solve it. In (1958) McCarthy posed the problem of getting a program with common sense in approximately these terms and suggested using sentences in an appropriate formal language to represent what the program knows. The advantage of representing information by sentences is that sentences have other sentences as logical consequences and the program can find consequences relevant to the goals at hand. Thus, representation of information by sentences allows the following:

1. A person can instruct the system without detailed knowledge of what sentences are already in the memory. Namely, the procedures for solving a problem using information in sentence form do not require that the information be in a particular order nor even a particular grouping of the information into sentences. All they require is that what to do is a logical consequence of the collection of sentences.

2. Similar considerations apply to information generated by the program itself.

3. Representing information by sentences seems to be the only clean way of separating that information which is common knowledge and so should be already in the system from information about a particular problem.

On the other hand, because each sentence has to carry with it much of its frame of reference, representation of information by sentences is very voluminous. It seems clear that other forms of information (e.g., tables) must also be used but the content of these other forms should be described by sentences.

In the last ten years considerable progress has been made in the use of the sentence representation. In the heuristic direction, theorem proving and problem solving programs based on J. Allen Robinson's resolution have been designed and continously improved. Cordell Green's QA3 (developed at the Stanford Research Institute and described in Reference 2,) and David Luckham's (this project) represent the present state of the art. Secondly, the theory of how to represent facts concerning causality, ability and knowledge for artificial intelligence purposes has been developed, mainly by McCarthy and his students. (McCarthy and Hayes, 1969) gives a good summary of where this work stands. The connection between this work and the subject of philosophical logic has been established, and some of the more recent efforts of philosophers to understand concepts of causality, ability and knowledge turn out to be of use in trying to make a computer use them. Another connection has been established with work in mathematical theory of computation. Namely, the results of McCarthy, Floyd and Manna about the correctness of computer programs can be used to make a system that finds strategies for achieving goals.

Our research work in representation by sentences is continuing along the following lines.

1. Improve the formal treatments in (McCarthy and Hayes, 1969) of the concepts of causality, ability and knowledge so as to get more realistic expressions of the "common knowledge" of these concepts.

2. Work on more realistic examples.

3. Translate the modal logic of the present formalism into first order logic so that present theorem proving and problem solving programs can be used.

4. Connect the work on representations with our work on representation of visual information where representation by sentences is rather clearly inappropriate.

Problems of representation also arise in our work with vision. In order to assemble an object out of parts or to drive a vehicle, the computer must have a suitable representation of the scene in its memory.

In the assembly or hand-eye case the representation of choice is a list of objects in the environment giving for each its shape (e.g., for objects with flat faces as a list of faces described by edges and vertices in turn) and its location. In the driving case we face another problem. Namely, most of the scene has to be dismissed as irrelevant. Thus a tree must be recognized and dismissed without generating a list of branches, twigs and leaves. That there are no obstacles on the road has to be verified with high reliability. Therefore, the representation in memory of a road scene must start with a division of the scan into "blobs" most of which are coarsely identified and dismissed while others are described in greater detail.

A project for such a description is being worked out along the following lines: The scene is divided into regions r in such a way as to minimize a quantity c that we call the complexity of the description. Letting R be the set of regions, we write

$$c = \sum_{r \in R} [a \times \text{perimeter}(r) + b \times \text{area}(r) \times \text{inhomogeneity}(r) + 1]$$

In the simplest case the inhomogeneity might be the variance of the brightness of the points in the region or the variance over the region of the vector of light intensities seen through three color filters. If the parameter b is small the optimal description will tend to have few regions so as to minimize the lengths of the boundaries. Thus a tree will appear as a single blob and the inhomogeneity resulting from the blue of sky, green of leaf, and brown of branches and trunk will be suffered. If the coefficient b is increased then the optimal description will separate out the trunks and use a longer boundary to separate the tree from the sky. A very large value of b would be required to make the system separate out each leaf and twig.

References

1. John McCarthy, "The Advice Taker" in Mechanisation of Thought Processes, Vol. 1, pp 77-84, Proc. Symposium, National Physical Laboratory, London, 1958. Reprinted in M. Minsky (ed), Semantic Information Processing, MIT Press, Cambridge, 1968.

2. Cordell Green, "The application of Theorem Proving to Question Answering Systems", Ph.D. Thesis in Electrical Engineering, Stanford University, 1969.

3. John McCarthy and Patrick Hayes, "Some Philosophical Problems from the Standpoint of Artificial Intelligence" in D. Michie (ed), Machine Intelligence 4, American Elsevier, New York, 1969.

2.2 Mathematical Theory of Computation

What are the fundamental properties and relations of algorithms, the data structures on which they operate, the programming languages in which they are expressed, the problems they are written to solve, and the computers that execute them? In our view, some of these fundamental properties and relations are:

1. The relations between the input and output information of an algorithm.

2. Whether the algorithm ever terminates for inputs satisfying certain conditions.

3. The equivalence of two algorithms.

4. Transformations of algorithms that preserve equivalence (and perhaps increase speed).

5. Relations between algorithms and the speed with which computers can carry them out.

6. The semantics of programming languages: i.e., the relation between the form of a program and properties of the algorithm it carries out.

The goal of our research has been a formal theory in which the relevant properties of actual algorithms can be stated and proved and the proofs checked by computer. Attainment of this goal would allow the replacement of debugging programs, (a never-ending process) by debugging proofs that programs have desired properties (a process that terminates conclusively when the proof-checker program accepts a proof that the given program has the desired properties).

These goals were first stated in (McCarthy, 1962) and progress was registered in that paper and in others by McCarthy, Painter, and

Kaplan (References 1-7). Results on correctness of programs and convergence in a different formalism were obtained by Floyd (1967), and Manna (1968). We should also mention the work of the IBM Vienna group in extending the methods and ideas of (McCarthy 1963, 1965) and applying them to PL/1. Now both Robert Floyd and Zohar Manna have joined the Stanford Computer Science Department and the Artificial Intelligence Project.

Within the last year Manna, McCarthy and Amir Pnueli have united the formalism of McCarthy with that of Floyd and have applied extensions of Manna's previous results to several concrete algorithms.

At present the part of mathematical theory of computation based on first order logic has reached a certain stage of completeness and Manna and McCarthy plan to write a book about it. Extensions of the formalism using set theory are being developed and a beginning has been made in extending Manna's method to parallel processes.

The first connections of this theory to the theorem proving research in artificial intelligence have been made by Green at Stanford and SRI. Green has shown how his theorem prover QA3 can be made to find a LISP program for sorting a list.

REFERENCES

1. John McCarthy, "A Basis for a Mathematical Theory of Computation", in P. Biaffort and D. Hershberg (eds), Computer Programming and Formal Systems, North-Holland, Amsterdam, 1963.

2. John McCarthy, "Towards a Mathematical Theory of Computation" in Proc. IFIP Congress 62, North-Holland, Amsterdam, 1963.

3. John McCarthy, "A Formal Description of a Subset of Algol" in T. Steele (ed), Formal Language Description Languages, North-Holland, Amsterdam, 1966.

4. John McCarthy, "Problems in the Theory of Computation", in Proc. IFIP Congress 65, Spartan, Washington D.C., 1965.

5. James Painter, "Semantic Correctness of a Compiler for an Algol-like Language", Ph.D. Thesis in Computer Science, Stanford University, 1967, (Memo AI-44).

6. Don Kaplan, "Regular Expressions and the Equivalence of Programs", Ph.D. Thesis in Computer Science (Memo AI-63), Stanford University, July 1968.

7. Don Kaplan, "Some Completeness Results in the Mathematical Theory of Computation", J. ACM, January, 1968.

8. Robert Floyd, "Assigning Meanings to Programs", in Proc. Symposia in Applied Math., Vol. XIX, American Mathematical Society, Providence, 1967.

9. Zohar Manna, "Termination of Algorithms", Ph.D. Thesis, Computer Science Department, Carnegie-Mellon University, Pittsburg, 1968.

10. Zohar Manna, "Properties of Programs and the First Order Predicate Calculus", J. ACM, April 1969.

11. Zohar Manna, "Formalization of Properties of Programs", J. System and Computer Sciences, May 1969.

12. Zohar Manna and Amir Pnueli, "Formalization of Properties of Recursively Defined Functions", Proc. ACM Symposium on Computing Theory, May 1969.

3. Visual Perception and Control

The overall goal of the hand-eye project is to design and implement a system which exhibits intelligent perceptual-motor behavior. An important subgoal is that the problems that arise in the design of system components be solved in ways which are sufficiently general to be scientifically interesting. Thus, for example, we have put considerable effort into understanding depth perception although the special environment we are using may allow for ad hoc solutions.

More specifically, our goals for the coming year include:

1) Theories and programs to solve basic recognition subtasks such as edge following, corner finding, object identification, etc.

2) A system which allows basic routines to be assembled into embodiments of perceptual and manipulative strategies.

3) A means of studying and using the interactions of the various basic processes.

To provide a sense of direction and to bound our aspirations, we proposed a class of tasks which we hope to have the hand-eye performing by 1971.

Two preliminary tasks are:

1) The ability to move the mechanical arm precisely by making use of visual feedback, and

2) The ability to pick up a specified object in a complicated scene and orient it.

These two tasks are prerequisites for our main task - the building of fairly complex constructions (castles) out of simple blocks. The blocks are restricted to being plane-bounded and convex. The castle might be explicitly described by a set of associations relating its sub-parts or we might simply be given one or more views of it.

One of the most interesting aspects of this task is the various levels of feedback which can be used in the building process. In some cases, one need only know that a block is still in place, whereupon tactile feedback is sufficient. If the situation is more critical one might visually determine the placement error and alter the remainder of the strategy accordingly. Finally, there is the possibility of adjusting the block, under visual control, until the error is sufficiently small [23].

The use of visual feedback in block stacking presents a rather different problem than those normally discussed in picture processing. The vision routine has the job of determining the accuracy with which some block was placed. The total scene may be very complicated and it would be absurd to perform a complete scene analysis. Furthermore, the properties of the blocks to be examined may be known in great detail and the vision routine would be able to take advantage of this fact. This example typifies the core problem: context - sensitive visual perception.

3.1 The Organization of a Visual Perception System

Perception, and most particularly visual perception, is a complex process requiring a system which is sensitive to all the various levels of detail of the environment. Furthermore, since the available data is potentially over whelming the system must have both the mechanisms and appropriate strategies to select what data are worthy of its attention and what level of detail is best suited to the current perceptual goal.

Our approach to the system design centers on two basic issues:

1) Levels of detail, and

2) strategies for attention.

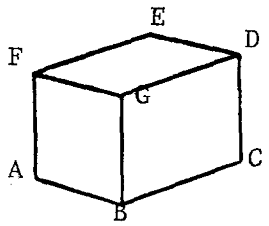
Data from a scene may be structured to varying degrees. At the lowest level lie the intensity and color of the light at a particular point in the visual field at a higher level are those objects in the visual scene which we dignify by the use of nouns; at a still higher level one notices interrelationships and relative motion between objects. At the highest level one is aware of the total situation -- as "Danger. Collision imminent." Ordinarily, we are conscious only of our perceptions of objects and situations, but the fact that we can learn to draw indicates that lower level details are perceived and can be made accessible to consciousness. It is curious that we must <u>learn</u> to draw -- as if the lower levels of visual patterns are coalesced into objects at a preconscious level. This notion gives rise to a simplified theory of perception held by many workers in perception and pattern recognition. The theory is embodied in a strategy of perception which places attention first at the lowest level of detail and then extracts successively higher levels until the organization of the entire scene is understood. Thus by processing intensity and color distributions one obtains texture, edges, and corners. From this information, regions are extracted and these in turn are associated into bodies. Then the bodies are identified as objects and their various interrelationships are derived. Thus:

Level: ^{1}points → ^{2}lines → ^{3}regions → ^{4}bodies → ^{5}objects → ^{6}scene

Essentially, all the early work on visual perception, including our own, proceeded along these lines. To some extent, the work of Guzman [11] on finding the distinct bodies in a perfect line drawing (level 2 ↠ level 4) had an undesirable effect on the field. Guzman's program was so successful that it sent people on a quest for the perfect line drawing program. Although we have had considerable success [7, 14] at generating line-drawings, it has become apparent that the strict bottom-to-top processing sequence is not optimal.

As an alternative to this linear approach, the model of vision which we find useful cooperativly involves analysis at various levels in an attempt to understand a scene. There is a large body of psychological evidence [4] indicating the dependence of perception upon global information and upon preconceived ideas. Although many of the well known optical illusions fall in this class, one can also show that there are simple scenes which are ambiguous in the absence of global inoformation, but are easily resolved in context.

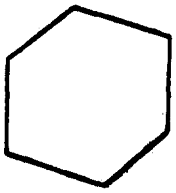
A most striking case of this is the ground plane assumption [15], which has become a cornerstone of all robot perceptual systems. From a monocular image it is impossible, in general, to calculate the distance of an object from the camera. If, however, the object is lying on a known plane (one whose transformation to image coordinates is available) then the depth of the object's base vertices is known. This particular piece of global information has been implicitly used for depth information, but has many other uses. Consider the following line drawing:



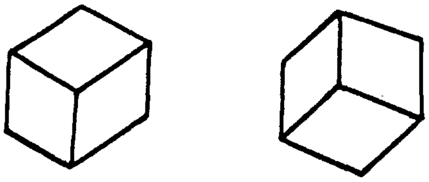
If one knew that this object were lying on the plane determined by ABC which is known, then one would know the projection of each point in the image onto the ABC plane. Each point, e.g., F must be on the line determined by its projection onto the ABC plane and the lens center. If the line AF is perpendicular to the plane we then know the length of AF.

Further, we can often determine whether or not AF is perpendicular to the plane from the information available. The lens center, point A and the projection of point F determine a plane, which contains the line AF. If this plane is perpendicular to ABC then the line AF is also, for objects which are at all regular [15]. If one knew the lengths of AF, BG, and CD and their angles with the ABC plane, then the coordinates of F, G, and D are computable and assuming F, G, D and E are in a plane is sufficient to determine E. More technically, the assumptions we have made allow only one degree of freedom in the choice of plane-bounded convex objects which could yield this image. Thus, the ground plane hypothesis plus some global regularity conditions allow for the complete description of an object from a single monocular view. Of course, these conditions may not hold, but we have some encouraging preliminary results in object recognition using these kinds of techniques.

A somewhat more basic problem arises in the consideration of the following image:



which might have come from, among other things:



The interior edges might very well be less distinct and be missed by the program which first tried to form a line drawing. At some higher perceptual level, a program could detect the ambiguity and attempt to find the interior edges. With the contextual information available, the system could then use highly specialized tests to determine the presence of an edge. Further, since the area involved is relatively small, it might also be reasonable to apply very sensitive general operations which are too costly to use on an entire scene.

In both examples we see how an organization which utilizes selective attention may facilitate perception. A vision system which worked strictly bottom-to-top would have no notion of attention. There would be a standard line finding operation, followed by an attempt to fit intersections, etc. In that kind of system there are inherent limitations [17] in balancing noise sensitivity against ability to perceive detail.

In addition to the enhanced perceptual ability attained by selective attention there are cost-benefit advantages. The vision programs will contain a wide variety of routines for analyzing various properties of pictures - any system which attempted to use all of them on each picture would be hopelessly inefficient.

The goal, then is to produce a flexible visual perception system capable of selective attention and of integrating information from all levels of perception. An obvious prerequisite for such a system is a monitor, language, and data structure capable of its support. Our proposed design is described in Section 3.2.

A second necessary ingredient of any cooperative system is a large set of flexible basic vision routines. Among the necessary functions are: reading raw data, changing the camera position and parameters, edge finding, corner fitting, region finding, analysis into distinct bodies, identification of particular objects, and complete scene analysis.

3.2 Hand-Eye Systems

Our first hand-eye system used many ad hoc solutions and was mainly concerned with the problems of combining the minimum necessary hardware and software components. This primitive, but complete system for block-stacking under visual control was completed in May, 1967 and has been described elsewhere [14]. The functional diagram of Figure 2 provides a sufficient description for our purposes. Our most recent work has involved the redesign of the system configuration and more careful study of each of the component programs.

The system configuration must be able to support a large number of program modules communicating with the world and with each other in complex ways. To achieve this capability we are undertaking a rather ambitious system-programming project including a submonitor, a high-level language, and a new data structure. The goal of this project is to produce a hand-eye laboratory in which it will be relativly easy to experiment with new ideas in perception, modeling, problem-solving and control. This laboratory will also, hopefully, provide a testing ground for many related artificial intelligence problems and should be relevant to the design of any large flexible real-time system.

The hand-eye laboratory will have to accomodate programs whose total size is several times the size of core memory. Further, as we have shown in Section 3.1, the order in which these programs are executed cannot be determined in advance. These programs must be able to communicate with each other and with a common global model which represents the system's knowledge of the world. Since many operations require moving physical devices (like the arm and camera) which entail long delays, we would like to allow parallel execution of hand-eye sub-programs. We are implementing

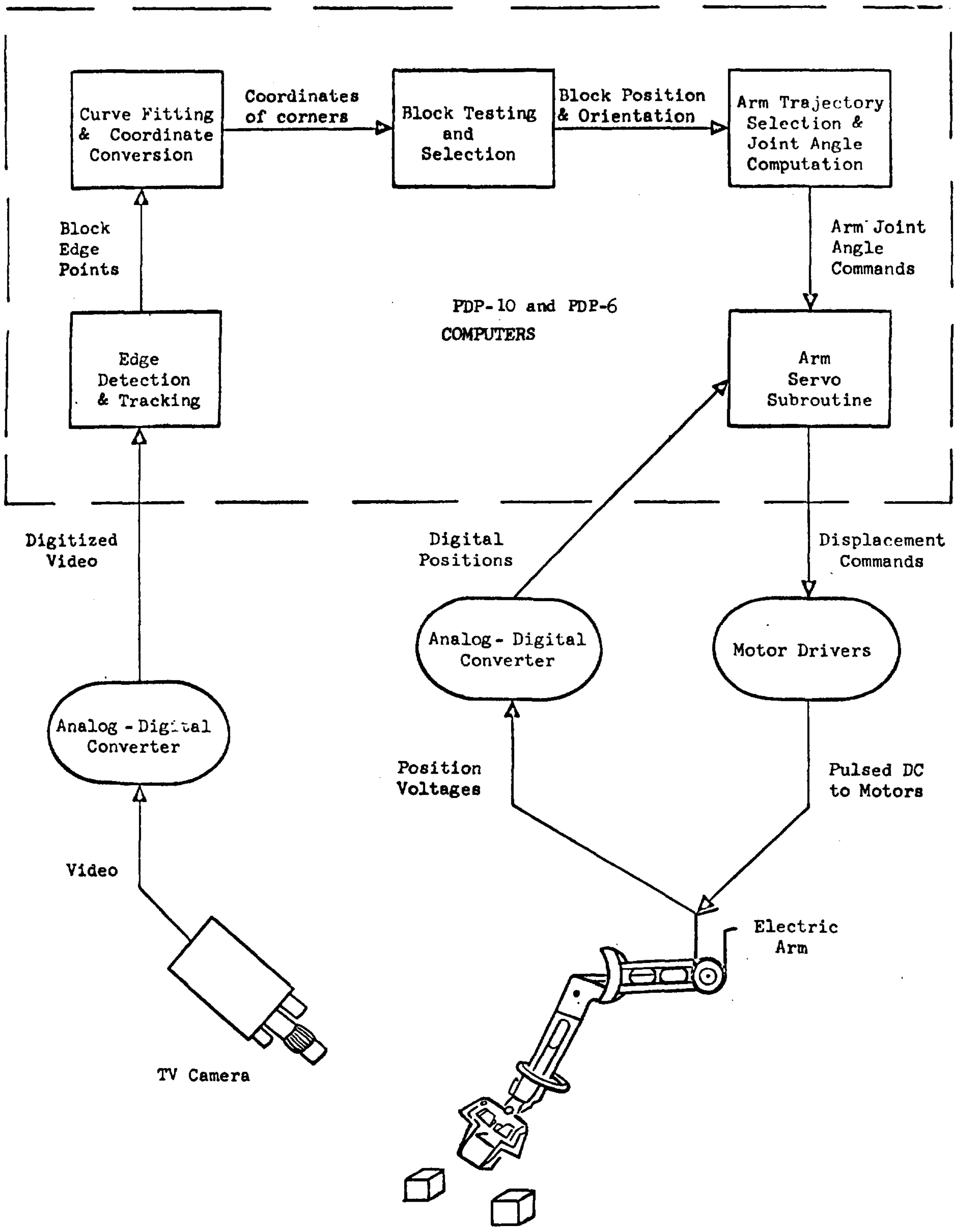

Curve Fitting & Coordinate Conversion
Coordinates of corners
Block Testing and Selection
Block Position & Orientation
Arm Trajectory Selection & Joint Angle Computation
Block Edge Points
Arm Joint Angle Commands
PDP-10 and PDP-6 COMPUTERS
Edge Detection & Tracking
Arm Servo Subroutine
Digitized Video
Digital Positions
Displacement Commands
Analog - Digital Converter
Motor Drivers
Analog - Digital Converter
Position Voltages
Pulsed DC to Motors
Video
Electric Arm
TV Camera

Figure 2 The Initial Block-Stacking System

a submonitor to perform these functions, handle messages, and carry out changes to the global model.

The language and data-structure designs are closely tied to the sub-monitor and to each other. The language will be an extension of our ALGOL Compiler [21] along the lines of the associative language, LEAP [3]. The central concept of LEAP and the underlying data structure is the association: attribute·object = value. The use of associations for world-modeling is described in detail in [9]. An important new concept of this version of LEAP is the use of local and global associative structures. The global structure contains the world model shared jointly among all programs working to analyze a scene. Every atomic object (item) is either local or global; the associative structure local to a subprogram may contain associations including global items, but not vice-versa. Any attempt to alter the global associative structure is trapped to the sub-monitor which determines when the alteration should be allowed. The language will contain primitives for local and global associations, message handling and interrupt processing.